# SPS-I Software Development

# Lessons Learned & Status

## EA-21 Data Summit

*14-15 October 1998*

## *SPS-I Software Development - Lessons Learned*
### *History*

- AMS Begins SPS-I Development Lifecycle to Replace APADE Interface Files to External Systems in July 1997
- AMS Completes Requirements Definition January 1998
- AMS Completes Development of SPS-I and Delivers to FMSO on June 30, 1998
- AMS Completes Installation of SPS-I at FMSO on July 2, 1998
- CDA Begins SPS-I Acceptance Test on July 7, 1998
- CDA is Continuing to Test SPS-I

## *SPS-I Software Development - Lessons Learned*
## *Current Status*

- SPS-I Acceptance Test Cases are 86% Complete (Oct. 1, 1998)
- 14% of Acceptance Test Cases Need to be Completed
- **???** Trouble Tickets Require Resolution

## *SPS-I Software Development - Lessons Learned*
## *Next Steps*

- CDA Completes Execution of Remaining Test Cases
- AMS Resolves Existing and Future Trouble Tickets
- AMS Delivers New Software Configuration to CDA
- CDA Completes Regression Testing of New Software Configuration
- CDA Certifies Acceptance of SPS-I Software
- AMS Installs SPS-I Software at FISC San Diego
- FISC San Diego Personnel Complete End-to-End Operational Test of SPS-I

## *SPS-I Software Development - Lessons Learned*
## *Next Steps - continued*

- Joint Interoperability Test Center (JITC) Personnel Perform Abbreviated Test of SPS-I
- JITC Completes Test Report and Submits to Major Automated Information System Review Council (MAISRC) for Review
- MAISRC Reviews Test Report and Provides Approval to Deploy SPS-I Software
- AMS Installs SPS-I Software at Applicable APADE Sites

## *SPS-I Software Development - Lessons Learned*
## *Overall Lessons*

- Ownership of Interface Software Through Partnering
  - Establish Rules of Engagement
  - Identify Roles and Responsibilities of Team Members
  - Create Integrated Product Team (IPT)
  - Ensure External System Participation
- Knowledge Sharing
  - Demonstration of Each System (SPS and Legacy)
  - Share Understanding of Business Processes Shared
  - Understand How Legacy System Interfaces with External System(s)

## *SPS-I Software Development - Lessons Learned*
## *Overall Lessons*

- Program Management
  - Require Narrative Project Plan Along with Schedule
  - Require Monthly Status Reports
  - Require In-Process Reviews (IPRs)
- Design Quality Into System From Requirements Definition Stage
  - Shortens Design, Development and Testing Periods
  - Eliminates Re-Design or Re-Coding
  - Maximizes Use of Control or Check Points

## *SPS-I Software Development - Lessons Learned*
## *Requirements Definition*

- Develop Memorandums of Agreement Between SPS-I and External System(s)
- Include Interviews of External System(s) Representatives in Requirements Definition Task
- Identify and Validate Functional Requirements and Business Rules as a Team
- Use Change Control Procedures for Updating Functional Requirements
- Include MQSeries (or other delivery solution) Architectures with System Documentation

## *SPS-I Software Development - Lessons Learned*
## *System Design and Development*

- Use Team Partners for Clarification of Issues When Necessary
- Include Team Partners in Design Reviews to Identify Discrepancies Before Development Begins
- Include Team Partners in Code Reviews
- Include Team Partners in Review of Unit and Integration Test Plans
- Include Team Partners in Review of Unit and Integration Test Results
- Include Team Partners in Integrated Development Testing at AMS Using Integrated Test Cases

## *SPS-I Software Development - Lessons Learned*
## *Acceptance Testing*

- Team Partners Establish Test Cases, Scenarios, or Scripts
- Use 3rd Party Facilitator to Mediate Test
- Initiate IPT Sessions When Issues or Pattern of Problems Arise
- Utilize On-Site Functional and Technical Support Provided by Vendor
- Ensure That Adequate Time is Scheduled for Resolution of Bug Fixes and Re-testing
- Hold Regular Status Meetings or Teleconferences to Resolve Issues
- Document Lessons Learned for Further Refinement of Test Activity
- Allocate Dedicated Help Desk Support Person for Trouble Tickets
- Adopt a “No Fault” Testing Philosophy to Maintain a Strong Working Relationship During Testing